

w05-Lab

# Conditional Control Statements

## Part II

Adapted for 204111  
by Areerat Trongratsameethong

## Logical Operators

- More complex conditions can be created using the logical operations
  - AND (&&)
  - OR (||)
  - NOT (!)
- When the && is used with two expressions, the condition is true only if both expressions are true by themselves

## Topics

- Logical Operator
- The if-else Chain
- The switch Statement
- Practice

## Logical Operators: &&

Table 4.3 The AND (&&) Operator

If expressionOne is:	And expressionTwo is:	Then, expressionOne && expressionTwo is:
true (that is, non-0)	true (that is, non-0)	true (1)
true (that is, non-0)	false (that is, 0)	false (0)
false (that is, 0)	true (that is, non-0)	false (0)
false (that is, 0)	false (that is, 0)	false (0)

<i>p</i>	<i>q</i>	<i>p &amp;&amp; q</i>
T	T	T
T	F	F
F	T	F
F	F	F

*p* = condition 1 (expression 1)

*q* = condition 2 (expression 2)

T = true

F = false

# Logical Operators: ||

**Table 4.4** The OR (||) Operator

If expressionOne is:	And expressionTwo is:	Then, expressionOne    expressionTwo is:
true (that is, non-0)	true (that is, non-0)	true (1)
true (that is, non-0)	false (that is, 0)	true (1)
false (that is, 0)	true (that is, non-0)	true (1)
false (that is, 0)	false (that is, 0)	false (0)

p	q	p    q
T	T	T
T	F	T
F	T	T
F	F	F

p = condition 1 (expression 1)

q = condition 2 (expression 2)

T = true

F = false

## Logical Operator Evaluation

```
int i = 15, j = 30;
double a = 12.0, b = 2.0, complete = 0.0;
```

Expression	Value	Interpretation
a > b	1	true
i == j    a < b    complete	0	false
a/b > 5 && i <= 20	1	true

# Logical Operators: !

**Table 4.5** The NOT (!) Operator

If expression is:	Then, !expression is:
true (that is, non-0)	false (0)
false (that is, 0)	true (1)

p	!p
T	F
F	T

p = condition (expression)

T = true

F = false

Example From: [http://www.kosbie.net/cmu/spring-13/15-112/handouts/notes-data-and-exprs.html#Short-circuit\\_evaluation](http://www.kosbie.net/cmu/spring-13/15-112/handouts/notes-data-and-exprs.html#Short-circuit_evaluation)

## Logical Operator Evaluation (2)

- The evaluation feature for the && and || operators that makes the evaluation of an expression stop as soon as it is determined that an expression is false is known as **short-circuit evaluation**

- Example:

```
int x = 0, y = 0, z;
if ((y != 0) && ((x/y) != 0)) z = 5;    // Works!
if (((x/y) != 0) && (y != 0)) z = 5;    // Crashes!
```

- Once again, using the "||" operator

```
int x = 0, y = 0, z;
if ((y == 0) || ((x/y) == 0)) z = 5;    // Works!
if (((x/y) == 0) || (y == 0)) z = 5;    // Crashes!
```

# Logical Operator Evaluation (3)

- Parentheses can be used to alter the assigned operator priority

```
(6 * 3 == 36 / 2) && (13 < 3 * 3 + 4) || !(6 - 2 < 5)
= (18 == 18)      && (13 < 9 + 4)      || !(4 < 5)
= 1                && (13 < 13)         || !1
= 1 && 0 && 0
= 1 && 0
= 0
```

# Logical Operator Precedence (2)

```
char key = 'm';
int i = 5, j = 7, k = 12;
double x = 22.5;
```

Expression	Equivalent Expression	Value	Interpretation
<code>i + 2 == k - 1</code>	<code>(i + 2) == (k - 1)</code>	0	false
<code>3 * i - j &lt; 22</code>	<code>((3 * i) - j) &lt; 22</code>	1	true
<code>i + 2 * j &gt; k</code>	<code>(i + (2 * j)) &gt; k</code>	1	true
<code>k + 3 &lt;= -j + 3 * i</code>	<code>(k + 3) &lt;= ((-j) + (3*i))</code>	0	false
<code>'a' + 1 == 'b'</code>	<code>('a' + 1) == 'b'</code>	1	true
<code>key - 1 &gt; 'p'</code>	<code>(key - 1) &gt; 'p'</code>	0	false
<code>key + 1 == 'n'</code>	<code>(key + 1) == 'n'</code>	1	true
<code>25 &gt;= x + 4.0</code>	<code>25 &gt;= (x + 4.0)</code>	0	false

# Logical Operator Precedence

Table 4.6 C Operators Listed from Highest Precedence to Lowest Precedence

Operator	Associativity
!, unary -, ++, --	right to left
*, /, %	left to right
+, -	left to right
<, <=, >, >=	left to right
==, !=	left to right
&&	left to right
	left to right
+=, -=, *=, /=	right to left

# Logical Operators: Example

- If and Logical Operator Example
- Love6 Game: Given two integer values, first and second. It will return True if:
  - Either one is 6
  - Their sum is 6
  - Their difference is 6

How to write condition?

# The if-else Chain

```

if (expression1)
    statement1;

else if (expression2)
    statement2;

else
    statement3;

```

```

if (expression1) {
    statement1;
    ...
} else if (expression2) {
    statement2;
    ...
} else {
    statement3;
    ...
}

```

# The if-else Chain (3)



```

1  #include <stdio.h>
2  int main()
3  {
4      char marcode;
5
6      printf("Enter a marital code: ");
7      scanf("%c", &marcode);
8
9      if (marcode == 'M')
10         printf("\nIndividual is married.\n");
11     else if (marcode == 'S')
12         printf("\nIndividual is single.\n");
13     else if (marcode == 'D')
14         printf("\nIndividual is divorced.\n");
15     else if (marcode == 'W')
16         printf("\nIndividual is widowed.\n");
17     else
18         printf("\nAn invalid code was entered.\n");
19
20     return 0;
21 }

```

# The if-else Chain (2)

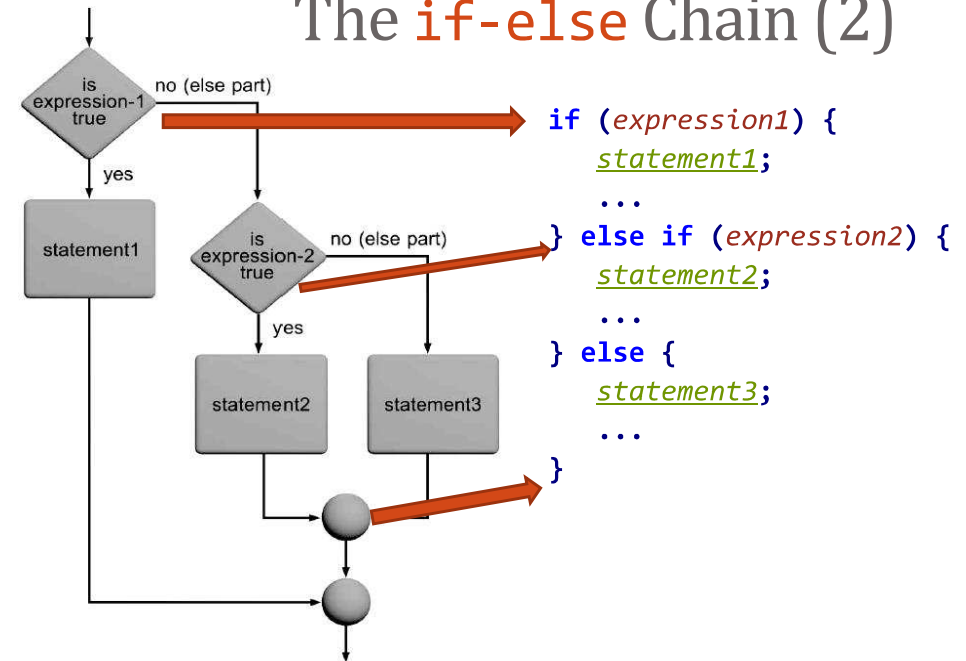


Figure 4.5 The if-else chain's flowchart

# The if-else Chain (4)

Monthly Sales	Income
greater than or equal to \$50,000	\$575 plus 16% of sales
less than \$50,000 but greater than or equal to \$40,000	\$550 plus 14% of sales
less than \$40,000 but greater than or equal to \$30,000	\$525 plus 12% of sales
less than \$30,000 but greater than or equal to \$20,000	\$500 plus 9% of sales
less than \$20,000 but greater than or equal to \$10,000	\$450 plus 5% of sales
less than \$10,000	\$400 plus 3% of sales

## The if-else Chain (5)

```

1 // ...
2
3 int main()
4 {
5     float monthlySales, income;
6
7     printf("Enter the value of monthly sales: ");
8     scanf("%f", &monthlySales);
9
10    if (monthlySales >= 50000.00)
11        income = 575.00 + .16 * monthlySales;
12    else if (monthlySales >= 40000.00)
13        income = 550.00 + .14 * monthlySales;
14    else if (monthlySales >= 30000.00)
15        income = 525.00 + .12 * monthlySales;
16    else if (monthlySales >= 20000.00)
17        income = 500.00 + .09 * monthlySales;
18    else if (monthlySales >= 10000.00)
19        income = 450.00 + .05 * monthlySales;
20    else
21        income = 400.00 + .03 * monthlySales;
22
23    printf("The income is $%.2f\n", income);
24
25    return 0;
26 }
27

```

17

## The if-else Chain (7)

```

char grade;
int score = 85;

if (score >= 90)
    grade = 'A';

else if (score >= 80)
    grade = 'B';

else if (score >= 70)
    grade = 'C';

else if (score >= 60)
    grade = 'D';

else
    grade = 'F';

printf("Score: %d, Grade: %c\n", score, grade);

```

19

## The if-else Chain (6)

```

char grade;
int score = 85;

if (score >= 90)
    grade = 'A';

if (score >= 80)
    grade = 'B';

if (score >= 70)
    grade = 'C';

if (score >= 60)
    grade = 'D';

else
    grade = 'F';

printf("Score: %d, Grade: %c\n", score, grade);

```

Correct?

18

## The switch Statement

- **switch statement:**
  - Allows a variable to be tested for **equality** against a list of values.
  - Each value is called a case
  - The variable being switched on is checked for each switch case.

Reference: [http://www.tutorialspoint.com/cprogramming/switch\\_statement\\_in\\_c.htm](http://www.tutorialspoint.com/cprogramming/switch_statement_in_c.htm)

20

## The switch Statement (2)

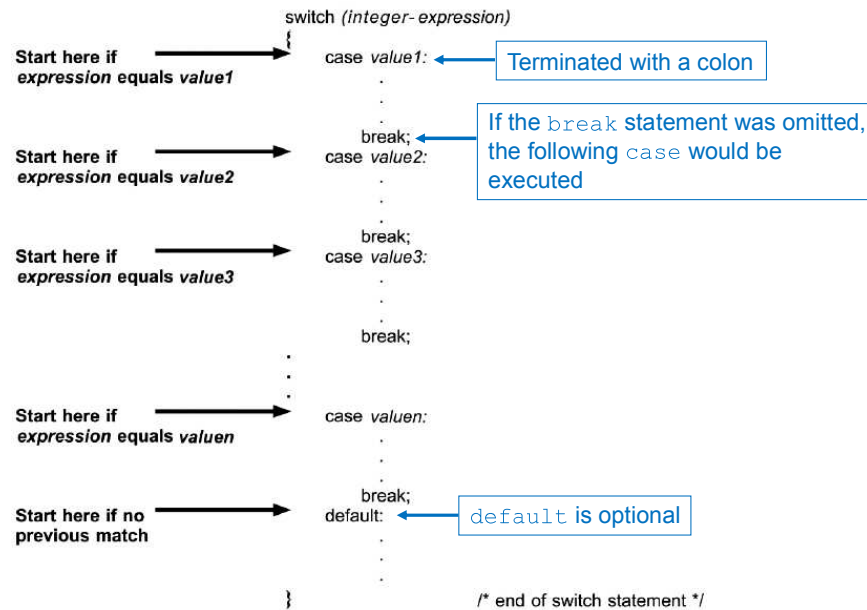


Figure 4.6 The expression determines the entry point

21

## The switch Statement (3)

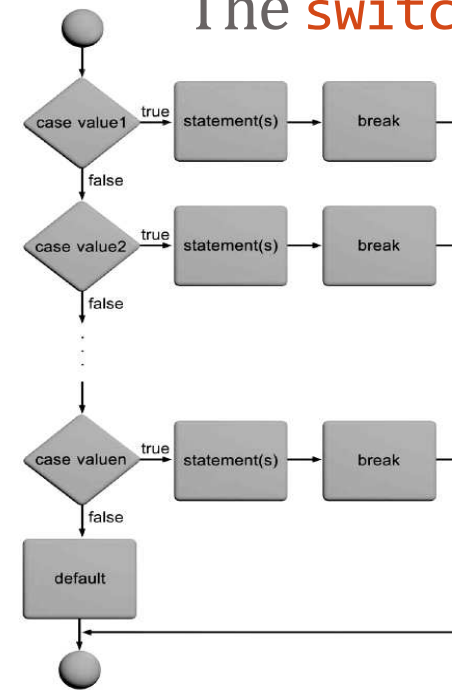


Figure 4.7 The switch flowchart

22

## The switch Statement (4)

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int opselect;
6      float fnum, snum;
7
8      printf("Please type in two numbers: ");
9      scanf("%f %f", &fnum, &snum);
10     printf("Enter a select code:");
11     printf("\n 1 for addition");
12     printf("\n 2 for multiplication");
13     printf("\n 3 for division : ");
14     scanf("%d", &opselect);
15
16     switch (opselect)
17     {
18         case 1:
19             printf("The sum of the numbers entered is %6.3f\n", fnum+snum);
20             break;
21         case 2:
22             printf("The product of the numbers entered is %6.3f\n", fnum*snum);
23             break;
24         case 3:
25             if (snum != 0.0)
26                 printf("The first number divided by the second is %6.3f\n", fnum/snum);
27             else
28                 printf("Division by zero is not allowed\n");
29             break; /* this break is optional */
30     } /* end of switch statement */
31
32     return 0;
33 } /* end of main() */
  
```

23

## The switch Statement (5)

input num		input num
if (num == 1) {	↔	switch (num) {
<u>expression1</u>		case 1:
...		<u>expression1</u>
break;		...
else if (num == 2) {	↔	break;
<u>expression2</u>		case 2:
...		<u>expression2</u>
break;		...
else if (num == 3) {	↔	break;
<u>expression3</u>		case 3:
...		<u>expression3</u>
break;		...
else {	↔	break;
<u>expression4</u>		default:
...		<u>expression4</u>
break;		...
}		}

24

```
# include <stdio.h>
```

```
int main()
{
    char o;
    float num1,num2;
```

```
printf("Select an operator either + or - or * or / \n");
scanf("%c",&o);
printf("Enter two operands: ");
scanf("%f%f",&num1,&num2);
switch(o)
{
    case '+':
        printf("%.1f + %.1f = %.1f",num1, num2, num1+num2);
        break;
    case '-':
        printf("%.1f - %.1f = %.1f",num1, num2, num1-num2);
        break;
    case '*':
        printf("%.1f * %.1f = %.1f",num1, num2, num1*num2);
        break;
    case '/':
        printf("%.1f / %.1f = %.1f",num1, num2, num1/num2);
        break;
    /* If operator is other than +, -, * or /, error message is shown */
    default:
        printf("Error! operator is not correct");
        break;
}
return 0;
```

ตัวอย่าง Output จากการ Run โปรแกรม

```
Enter operator either + or - or * or /
*
Enter two operands: 2.3 4.5
2.3 * 4.5 = 10.3
```

```
/* C program to demonstrate the working of switch...case statement */
```

```
/* C Program to create a simple calculator for addition, subtraction, multiplication and division */
```

204111: Fundamentals of Computer Science

## The switch Statement (8)

```
#include <stdio.h>
```

```
int main()
{
    char c;
    int count_a = 0, count_b = 0, other = 0;
```

```
printf("Input a character: ");
scanf("%c",&c);
```

```
switch (c) {
    case 'A':
    case 'a':
        count_a++;
        break;
    case 'B':
    case 'b':
        count_b++;
        break;
    default :
        other++;
}
```

```
printf("Input character is %c, count_a = %d, count_b = %d,
        other = %d", c, count_a, count_b, other);
return 0;
```

```
}
```

```
case 'A':
case 'a':
is similar to
if (c == 'A' || c == 'a')
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 1

```
Input a character: A
Input character is A, count_a = 1, count_b = 0, other = 0
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 2

```
Input a character: B
Input character is B, count_a = 0, count_b = 1, other = 0
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 3

```
Input a character: c
Input character is c, count_a = 0, count_b = 0, other = 1
```

## The switch Statement (7)

```
#include <stdio.h>
```

```
int main()
{
    char c;
    int capital_a = 0, letter_a = 0, other = 0;
```

```
printf("Input a character:");
scanf("%c", &c);
```

```
switch(c) {
    case 'A':
        capital_a++;
    case 'a':
        letter_a++;
    default :
        other++;
}
```

```
printf("Input character is %c, capital_a = %d, letter_a = %d,
        other = %d", c, capital_a, letter_a, other);
return 0;
```

```
}
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 1

```
Input a character: A
Input character is A, capital_a = 1, letter_a = 1, other = 1
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 2

```
Input a character: a
Input character is a, capital_a = 0, letter_a = 1, other = 1
```

ตัวอย่าง Output จากการ Run โปรแกรม ครั้งที่ 3

```
Input a character: c
Input character is c, capital_a = 0, letter_a = 0, other = 1
```

204111: Fundamentals of Computer Science

## Practice 1

1. แก้ปัญหาการทอนเหรียญ (10, 5, 2, 1) กรณีนี้นับ 50 โดยกำหนดให้เงินที่จ่ายต้องอยู่ในช่วง 1 ถึง 50 บาทเท่านั้น (ไม่มีเศษสตางค์) โดยที่การทอนต้องทอนเหรียญเรียงตามค่ามากที่สุดก่อน เช่น

- ราคาซื้อ 12 บาท ต้องทอน 38 บาท

- 10 บาท 3 เหรียญ
- 5 บาท 1 เหรียญ
- 2 บาท 1 เหรียญ
- 1 บาท 1 เหรียญ

1.1 เขียน Pseudocode และ Flowchart

1.2 ออกแบบ Input และ Output สำหรับใช้ทดสอบโปรแกรม

1.3 เขียนโปรแกรมภาษา C ตามที่ออกแบบไว้ในข้อ 1.1



## Practice 2

2. แก้ปัญหาการตัดเกรด โดยมีรายละเอียดของการตัดเกรดดังแสดงในตารางด้านล่าง ทั้งนี้คะแนนเป็นเลขทศนิยม

คะแนน	เกรด
ตั้งแต่ 80 ขึ้นไป	A
ต่ำกว่า 80 แต่ไม่เกิน 70	B
ต่ำกว่า 70 แต่ไม่เกิน 60	C
ต่ำกว่า 60 แต่ไม่เกิน 50	D
ต่ำกว่า 50	F

2.1 เขียน Pseudocode และ Flowchart

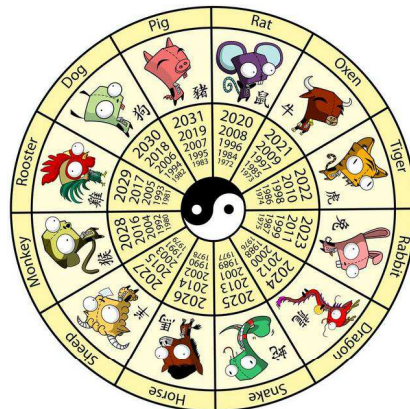
2.2 ออกแบบ Input และ Output สำหรับใช้ทดสอบโปรแกรม

2.3 เขียนโปรแกรมภาษา C ตามที่ออกแบบไว้ในข้อ 2.1

## Practice 4.1

- ให้เขียนผังงานและรหัสเทียมของโปรแกรม เพื่อแสดงวิธีการคำนวณการหาปีนักษัตร (Chinese Zodiac Year) จากปี ค.ศ. โดย

- ปีนักษัตร เป็นปีที่ใช้สัญลักษณ์เป็นสัตว์ 12 ตัว ประกอบด้วย Rat, Ox, Tiger, Rabbit, Dragon, Snake, Horse, Goat, Monkey, Rooster, Dog, Pig
- จะมีการวน แสดงตามปีเป็นรอบ ๆ รอบละ 1 ปี



## Practice 3

3. ตรวจสอบอักขระที่ผู้ใช้นำเข้า โดยที่ตรวจสอบว่าเป็นเลข 0-9 หรือ ตัวอักษรภาษาอังกฤษ หากผู้ใช้นำเข้าตัวอักษร ให้ตรวจสอบเพิ่มเติมว่าเป็นสระหรือไม่

3.1 เขียน Pseudocode และ Flowchart

3.2 ออกแบบ Input และ Output สำหรับใช้ทดสอบโปรแกรม

3.3 เขียนโปรแกรมภาษา C ตามที่ออกแบบไว้ในข้อ 3.1

## ตัวอย่างการเทียบปีนักษัตรจากปี ค.ศ.

ปี ค.ศ.	ปีนักษัตร
1992	Monkey (วอก)
1993	Rooster (ระกา)
1994	Dog (จอ)
1995	Pig (ขุน)
1996	Rat (ชาวด)
1997	Ox (ฉลู)
1998	Tiger (ขาล)
1999	Rabbit (เถาะ)
2000	Dragon (มะโรง)
2001	SNAKE (มะเส็ง)
2002	Horse (มะเมีย)
2003	Goat (มะแม)



## Practice 4.2

- ให้เขียนผังงานและรหัสเทียมของโปรแกรม เพื่อแสดงวิธีการคำนวณการหาธาตุประจำปี (Elements) โดย
- ธาตุประจำปีเกิด จะประกอบด้วย 5 ธาตุ ได้แก่ **Water** **Wood** **Fire** **Earth** **Metal** โดยจะเวียนครบละ 2 ปี
- (hint: สามารถดูจากตัวหลังของ ปี ค.ศ.)
  - 0-1 = Metal
  - 2-3 = Water
  - 4-5 = Wood
  - 6-7 = Fire
  - 8-9 = Earth
- ดังนั้นแต่ละปีเกิดจะถูกกำหนดโดยธาตุและนักษัตรเช่น
- มังกรทอง หนูไฟ ม้าน้ำ เป็นต้น

